

Fast Unsupervised Learning for Visual Pattern Recognition using Spike Timing Dependent Plasticity

Daqi Liu^a, Shigang Yue^a

^a*School of Computer Science, University of Lincoln, Brayford Pool, Lincoln, LN6 7TS*

Abstract

Real-time learning needs algorithms operating in a fast speed comparable to human or animal, however this is a huge challenge in processing visual inputs. Research shows a biological brain can process complicated real-life recognition scenarios at milliseconds scale. Inspired by biological system, in this paper, we proposed a novel real-time learning method by combining the spike timing-based feed-forward spiking neural network (SNN) and the fast unsupervised spike timing dependent plasticity learning method with dynamic post-synaptic thresholds. Fast cross-validated experiments using MNIST database showed the high efficiency of the proposed method at an acceptable accuracy.

Keywords: SNN, rank order coding, unsupervised, STDP, visual pattern recognition, fast learning

1. Introduction

For human and most of the animal species, reliable and fast visual pattern recognition is vital for their survival. In most cases, new visual pattern should be learned in a limited time window to adapt to new environments or changes promptly. Real-time learning has to learn/perform with limited samples often in real time, with no opportunity to learn the whole training samples/database. Similar learning situation applies to many machine learning scenarios - a rescue robot needs to learn to recognize individuals on the spot, an identification or

*Corresponding author

Email addresses: dliu@lincoln.ac.uk (Daqi Liu), syue@lincoln.ac.uk (Shigang Yue)

human recognizing system needs to cope with new criminals in any public areas, an health-care intelligent machine need to learn to cope with new patients quickly with limited information, all of which need to learn in a fast speed in real time.

To enable a machine to learn visual patterns in real time proposes a huge challenge - the algorithms underlying should process a large volume of visual data in an extremely short period of time, which has been proved difficult. However, it is also obvious that a biological brain could cope with these large volume of visual data effortlessly in real time. A human brain may contain more than 10 billion densely packed neurons that are connected to an intricate network with numerous spikes are emitted in each millisecond. Although, the mechanism of how these spikes are generated and processed is still an open question - this has not prevented researchers from proposing biological plausible methods for pattern recognition, as briefed below.

Various coding schemes have been proposed during the last several decades, such as spike rate-based coding scheme [1],[2], and spike timing-based coding scheme [3], [4]. A human brain can recognize objects in a few tens of milliseconds in a very complicated real-life scenarios [5]. In such a short time window, it is almost impossible for rate-based spiking neural network to generate meaningful spiking rate. Because of the existence of repeating spatiotemporal spiking patterns with millisecond precision, both in vitro and in vivo [6], rate-based spiking neural network would not be able to discriminate the repeating spiking patterns from the distractor. On the other hand, the spike timing-based SNN works at short time window and can extract a repeating pattern with appropriate learning rule [6]. Furthermore, a spiking pattern itself conveys significant structural information, which cannot be represented by spiking rate alone. Therefore, spike timing-based coding scheme, e.g. Rank Order Coding (ROC) scheme will be used in this study to translate the features to spiking patterns for fast visual information processing. These spiking patterns convey unique spatiotemporal structural information and can be used to distinguish different input images.

To learn these spiking patterns, spike timing dependent plasticity (STDP)

[7],[8],[9],[10],[11], a biological process that adjusts the efficacy of synaptic connections based on the relative timing of post-synaptic spikes and its input presynaptic spikes, is one of the most biological plausible learning rule. Like Hebb’s postulate [12], it emphasizes “Cells that fire together, wire together”. As an unsupervised learning rule, STDP does not need prior information or teaching
 45 signal in learning. It will adaptively change the synaptic efficacy and try to extract the most notable spiking pattern.

There are a few research papers on SNN and spike timing-based coding scheme for visual pattern recognition [13], [14]. Inspired by HMAX model [15] which consists of four layers (S1-C1-S2-C2) to simulate ventral stream (V1-V2-V4-IT), Thorpe et. al [13] have investigated the learning of C1 to S2 synaptic connections through STDP and suggested that temporal coding may be a key to understand the phenomenal processing speed achieved by the visual system. However, in [13] STDP was not used for spiking pattern recognition but local feature extraction. In the paper [14], the authors proposed a novel SNN with supervised learning rule and temporal coding scheme to generate the spike pattern.
 55 The proposed SNN and its supervised learning rule achieved relatively good classification rate when conducting cross-validation experiments on the MNIST database. Such supervised learning rule needs prior knowledge before learning - in many cases, this prior knowledge is hard to obtain.

To address the fast learning issues with biological plausible approaches, we propose a novel method taking advantages of SNN and spiking timing-based coding scheme. Different to [13], in our method, S1 and C1 are only for feature extraction; features are translated to spiking pattern from C1; STDP is used after spiking encoding layer for pattern recognition. To further speed up the leaning process, we only use S1 and C1 to extract visual features. Unsupervised
 65 learning rule is employed in the proposed method to make it more practical. Dynamic threshold is introduced to guaranty each training sample can be fully exploited in learning. Fast cross-validated experiments using MNIST database are carried out to prove the efficiency and accuracy of the proposed method.

70 The layout of this paper can be summarized as follows: Section II introduces

the framework of the proposed feed-forward spiking neural network. Neuron model and STDP learning rule are presented in Section III. Section IV depicts the experimental results, along with the analysis about the results. Finally, Section V summarizes this paper, discussing the advantages and limitations of the proposed method.

2. Framework of the proposed feed-forward spiking neural network

The whole framework of the proposed feed-forward spiking neural network will be explained in this section. In real world, the visual pattern recognition scenario often contains vast data dimensions and exists significant variability in terms of inter-class and intra-class. The first step in almost all visual pattern recognition tasks is to reduce data dimensionality, which means more generalized features need to be generated firstly. The generated features should contain the most distinguishable and unchangeable characteristics of the original input image [16], [17]. Until now, the input data are still analog values, which need to be transferred to spike trains for further learning. We use ROC to encode the analog values to spiking trains in this study.

The structure of the proposed feed-forward SNN can be summarized as following: feature extracting layer, spiking encoding layer and output layer. Fig.1 shows the framework of the proposed spike timing-based feed-forward SNN. Feature extracting layer computes C1 features with different scales and directions from input images. ROC scheme transfers those C1 features into spike trains within the spiking encoding layer. Each input image has its own corresponding spike pattern after those two layers. In the output layer, STDP learning rule and winner-take-all strategy have been used to train the synaptic efficacy matrix with specific selectivity to the input image. Notably, there is only one neuron (corresponding to specific class) within each output map in this paper. Below is the details of the three layers.

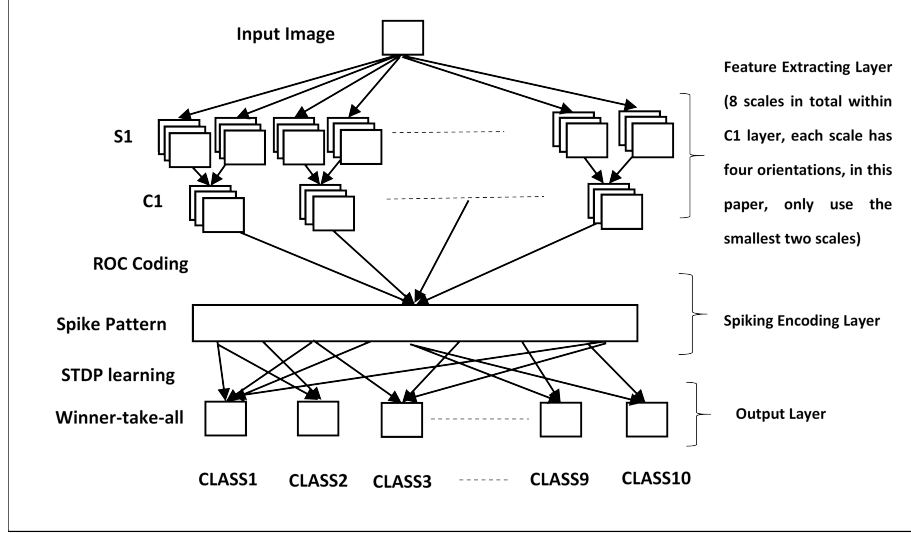


Figure 1: The framework of the proposed timing-based feed-forward spiking neural network.

2.1. Feature Extracting Layer

It has been shown that visual processing is hierarchical, aiming to build an invariance to position and scale first and then to viewpoint and other transformations [18]. In HMAX model [15],[19], the authors proposed a hierarchical system that closely follows the organization of visual cortex and thus built an increasingly complex and invariant feature representation by alternating between a template matching and a maximum pooling operation. Basically, this hierarchical system can be divided into 4 layers: S1 layer, C1 layer, S2 layer, C2 layer. The simple S units within S1 and S2 combine their inputs with a bell-shaped tuning function to increase selectivity. The complex C units within C1 and C2 pool their inputs through a maximum operation, thereby increasing invariance. C1 units and C2 units mimic the complex cells in V1 and cells in V4. Fig.2 illustrates the template matching and the max pooling layers used in HMAX model.

In this paper, for the sake of efficiency and simplicity, we used first two layers of HMAX model to extract the expected features, which includes S1 and C1 features. The features extracted from C1 layer can be used to mimic the

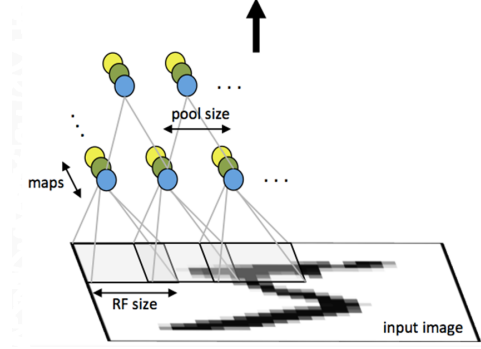


Figure 2: The template matching and the max pooling layers within HAMX model adopted in this paper. Units with the same color have tied weights and units of different color represents different filter maps.

115 complex cells in V1, thus they convey a relatively local invariance. Specifically, S1 features can be generated after applying Gabor filters with vary scales and orientations to the input image, which correspond to the classical simple cells in the primary visual cortex. It has been shown that Gabor response ($F_{(x,y)}^{\sigma,\theta}$) can provide a good model of cortical simple cell receptive fields (Fig.3 shows Gabor
120 filter kernels with different scales and orientations), which can be described as follows:

$$F_{(x,y)}^{\sigma,\theta} = \exp\left(-\frac{(x_0^2 + \gamma^2 y_0^2)}{2\sigma^2}\right) \times \cos\left(\frac{2\pi}{\lambda} x_0\right), \text{ s.t.} \quad (1)$$

$$x_0 = x \cos\theta + y \sin\theta; \quad y_0 = -x \sin\theta + y \cos\theta \quad (2)$$

where x and y describes abscissa and ordinate of the input image, respectively. x_0 and y_0 represents abscissa and ordinate after rotating θ , respectively. γ represents aspect ratio, θ depicts the orientation, σ is the effective width and λ
125 the wavelength. In this paper, we choose the same parameters settings as the HMAX model that is using a range of sizes from 7×7 pixels to 37×37 form the pyramid of scales, and four orientations ($0^\circ, 45^\circ, 90^\circ, 135^\circ$) have been used. Notably, those S1 features have been normalized to a predefined range $[-1, 1]$ so that input images with the same contrast will generate same S1 features.

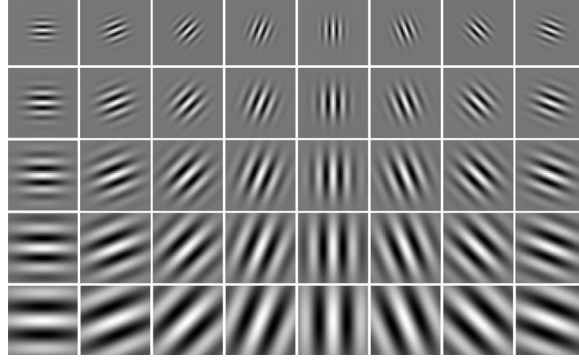


Figure 3: Gabor filter kernels with different scales and orientations.

130 C1 unit pool over retinotopically organized afferent S1 units from the previous S1 layer with the same orientation and from the same scale band. It corresponds to the cortical complex cells in V1, which convey certain invariance to local transformation. The vital part of C1 layer is the max pooling operation, which increases the tolerance to transformation from S1 layer to C1 layer.

135 Basically, the response $r_{(x,y)}^{\sigma,\theta}$ of a complex C1 unit corresponds to the maximum response of its m afferents $\left(F_{(x_1,y_1)}^{\sigma,\theta}, \dots, F_{(x_m,y_m)}^{\sigma,\theta}\right)$ from the previous S1 layer with two adjacent scales:

$$r_{(x,y)}^{\sigma,\theta} = \max_{j=1 \dots m} F_{(x_j,y_j)}^{\sigma,\theta} \quad (3)$$

In this paper, the first two smallest scales within C1 layer have been used to generate spike pattern. Fig.4 shows the procedure of generating spike pattern with the first smallest scale within C1 layer. The whole procedure can be summarized as follows: Firstly, the Gabor filter equations described in formula (1) and (2) have been used to generate S1 layer with adjacent scales and four different orientations. The parameter settings for the Gabor filters are the same as HMAX model. Secondly, for each orientation of the S1 layer, a local sliding window with the size of 8×8 has been applied to generate the C1 feature - notably, there are overlaps between two adjacent sliding windows and the overlapping size is 4×4 . Specifically, for each orientation, all 64 S1 features

140

145

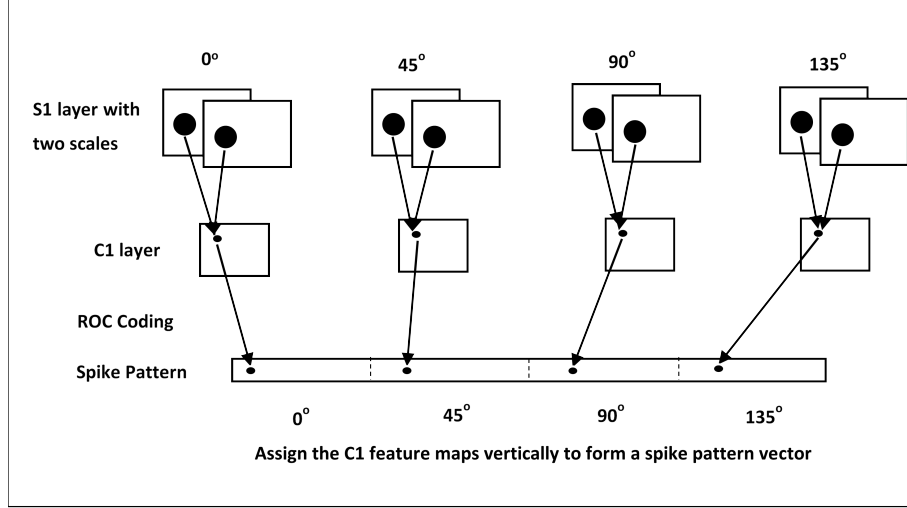


Figure 4: The generating procedure of spike pattern with the first smallest scale within C1 layer as example.

within each scale (2×64 S1 features in total) have been extracted to compute their maximum value. This maximum value will replace the two sliding window of S1 layer with two different scales. For the second smallest scale, the sliding windows size is 10×10 and the overlapping size is 5×5 . Throughout the feature extracting layer, only the most significant S1 feature within the corresponding sliding window has been selected and all others have been discarded. Such max pooling operation cannot only ensure the generated C1 feature having certain local invariance but also deduct the dimension of the whole data set.

It has been shown that these cortical complex cells tend to have larger receptive fields compared with simple cells within V1 [15]. For each orientation, C1 unit pool over two S1 maps with adjacent filter sizes. Those maps have the same dimensionality but they are the products of different filters. By sub-sampling with local cells with predefined sizes, C1 units takes the maximum response from the associated cell grid. Thereby, the dimension has been reduced with this max pooling operation. The bigger the cell grid takes, the lesser the dimensionality of C1 maps will be. Since S1 features have been normalized to $[-1, 1]$,

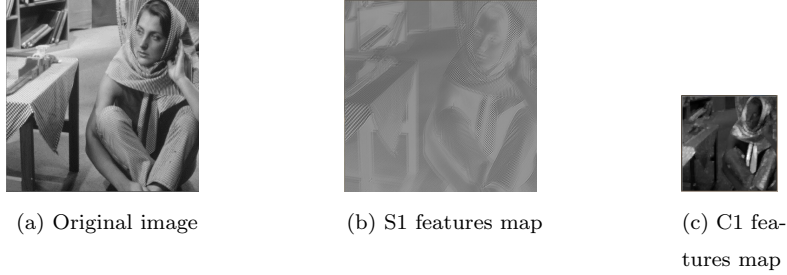


Figure 5: One input image and its associated S1 and C1 features maps (C1 map has been enlarged for better viewing).

C1 features will naturally have the range $[0, 1]$. By doing this, one can easily
 165 design the linear transformation strategy used in ROC scheme. Fig.5 shows the
 Barbara input image and its S1 and C1 features maps (for the sake of simplicity,
 only one S1 and C1 map shown here).

In a word, the feature extracting layer used in this paper mimic the cortical
 simple cells with S1 units and complex cells in V1 with C1 units. Template
 170 matching operation used in S1 layer generates orientation edge packages with
 certain selectivity, while max pooling operation in C1 layer achieves dimension-
 ality reduction and invariance to local transformation.

2.2. Spiking Encoding Layer

There are several commonly used spiking coding schemes: rate coding, tem-
 175 poral coding and population coding. Each coding scheme has its own advantages
 and drawbacks. Rate coding is the most well-known coding scheme, which con-
 sider the spiking rate conveys almost all information about the spike trains.
 Lots of works has been done with this coding scheme [1],[2],[20],[21]. However,
 several studies have shown that this rate coding scheme cannot deal with com-
 180 plicated visual pattern recognition tasks within a relatively short time window
 [22],[23],[24]. From the theoretical point of view, population coding [25],[26] is
 one of a few mathematically well-formulated problems in neuroscience. It repre-
 sents stimuli by using the joint activities of a number of neurons and thus each

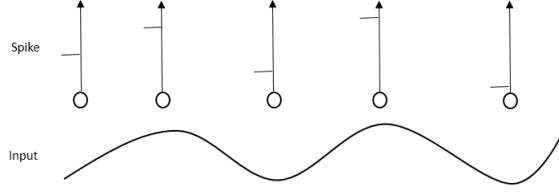


Figure 6: Rank order coding scheme diagram.

neuron has a distribution of response over some set of inputs. However, the
 185 structural complexity and the learning cost of this coding scheme seem to be
 quite high. Temporal coding scheme considers the specific precise spiking tim-
 ing conveys almost all information about the input spike trains. Several studies
 have found that the temporal resolution of the neural code is on a millisecond
 time scale, indicating that precise spike timing is a significant element in neural
 190 coding [27]. High-frequency fluctuation of firing-rates exhibited in neurons were
 considered as noise in rate coding scheme, however, in temporal coding scheme,
 they actually convey vital information about the input spike trains.

In this paper, rank order coding scheme [28],[29],[30], a time-to-first-spike
 coding scheme (one kind of temporal coding scheme), has been used to generate
 195 spike trains from the features extracted in the previous layer. Fig.6 shows the
 rank order coding scheme diagram. It can be seen that this kind of scheme only
 generate one spike after the corresponding unit receiving the input. The delay
 of the spiking timing is a monotonically decreasing function of the input analog
 value. Thus, the maximum analog input value corresponds to the minimum
 200 spiking timing delay. Pixels with less Input analog values will not generate spikes
 at all since their spiking timings have already exceeded a predefined time-window
 for spiking encoding layer (50 *ms* for this paper). Through such coding scheme,
 only those units with more significant C1 features will be generating spikes.
 Notably, only one spike will be generated for each unit in rank order coding
 205 scheme. Such coding scheme is intuitive yet powerful. Given the reference
 timestamp (the beginning time of the encoding procedure), it transforms each
 analog value into corresponding relative spike time associated with the reference

timestamp. Either the onset of external stimuli or the background oscillation can be considered as the reference timestamp. Although sometimes it is hard to find
210 these kinds of reference timestamps during the real world learning procedure, it is intuitive to use the onset of C1 features as the reference timestamp in the proposed spiking neural network. Another drawback of the classical rank order coding scheme is that its distinguishability (or selectivity) remains at a relatively low level if using the traditional relative coding method [29].

215 In this paper, we linearly modified the original rank order coding scheme so that absolute spiking timing instead of relative spiking timing [29] has been generated. For one specific feature response (depicted as r) within C1 layer, the corresponding spiking timing (s) can be computed as follows:

$$s = p * (\max(r) - r) \quad (4)$$

where $\max(r)$ is the maximum value of all related C1 features in the receptive
220 filed and p is a positive constant within the range from 0 to 1 (p takes 0.25 in this paper). Fig.4 also shows the details of the spiking encoding layer. Given the C1 maps with all four orientations, the formula (4) has been used to compute the exact spiking timing of the corresponding C1 feature. Notably, by vertically assigning the C1 map, each C1 map with certain orientation has been
225 transformed to a horizontal vector with the same orientation. Only one scale of C1 layer has been shown in Fig.4. In fact, we use the two smallest scales within C1 layer in the proposed method.

Through spiking encoding layer, C1 features will be transformed into spike trains. Fig.7 shows one input image and its spike pattern after processing with
230 the first two layers. Such spike trains can be considered as a spike pattern. Therefore, each input image will generate its own unique spike pattern through the first two layers. Such spike pattern contains specific spatiotemporal structural information about its input image and thus the selectivity to this specific input image has been emerged. Ideally, at least from the learning method's
235 perspective, one can expect that those spike patterns generated from the same

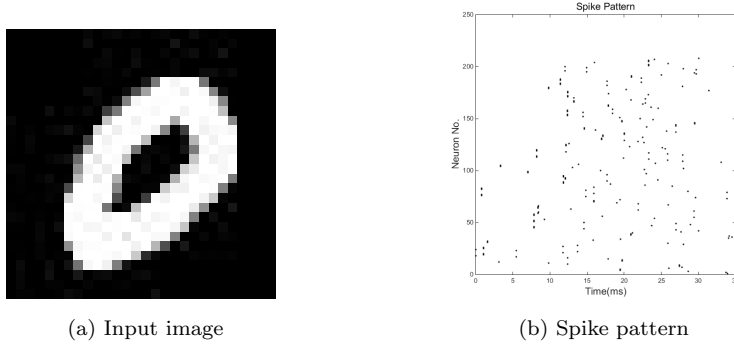


Figure 7: Input image and its spike pattern generated from the first two layers.

class would somehow looks similar to each other, while spike patterns from vary classes would be significantly different.

2.3. Output Layer

This output layer includes several neurons and the total number of neurons
 240 is the same as the total classes. The neurons within spiking encoding layer and
 output layer are fully connected so that each output neuron receive synaptic
 connections from all the neurons within spiking encoding layer. The output
 layer uses winner-take-all strategy so that the first fired neuron will strongly
 depress the rest neurons within the output layer from firing spikes and thus the
 245 input image will be considered as the class associated with the fired neuron. So
 there are lateral depression connections appears in the last layer.

The output layer is the only learning layer in the proposed feed-forward
 spiking neural network. From above two layers, the spike pattern associated
 with the input image will be generated and such spike pattern conveys cer-
 250 tain selectivity to its input image. Specifically, the spatiotemporal information
 embedded within the spike pattern plays an important role in defining such se-
 lectivity. The learning method within output layer should fully investigate such
 spatiotemporal information and thus use the learning results to distinguish the
 testing samples.

255 Spike timing dependent plasticity (STDP) learning rule is employed to the
output layer and it will dynamically changes the synaptic efficacy according to
the learning window. Eventually, the synaptic efficacy matrix will be stabilized
and thus the selectivity will be emerged after the learning procedure. Notably,
the next input image should feed into the feed-forward spiking neural network
260 only until the current input image has been successfully trained or tested. After
successfully learning the current input image, the intermediate variables gen-
erated during the training procedure will be reset to default values, except for
the learning efficacy matrix, which would be described in the following section.
This learning efficacy matrix would be updated each time until the very last
265 training image been feeding into the spiking neural network.

As described in the above sections, the whole visual pattern recognition
framework contains two important parts: *spike pattern generating* and *spike*
pattern learning. According to modified rank order coding scheme, the former
one generates *spike pattern* based on the C1 features. While the latter one uses
270 unsupervised STDP learning rule to learn the generated *spike pattern* and thus
obtained the final synaptic efficacy matrix with certain selectivity.

Unlike [13], in our method, STDP is used after spiking encoding layer for
pattern recognition and C1 features are translated to spiking patterns (S1 and
C1 are only for feature extraction).

275 3. Neuron Model and STDP Learning Rule

Neuron model represents the conduct principle of a spiking neuron. Leaky
integrate-and-fire model (LIF) and Spike Response Model (SRM) are the most
commonly used neuron models [31] in modern spiking neural networks. The
latter one tries to mimic the post-synaptic potential time course to an incoming
280 spike, which can also be considered as a generalization of LIF model. In this
paper, leaky integrate-and-fire neuron model has been used, as described in
details in the following sub section.

3.1. Neuron Model

Leaky integrate-and-fire neuron model acts as a coincidence detector and the causality between local spikes has been emphasized. When the postsynaptic neuron receives a spike from its presynaptic neuron, the responding postsynaptic potential (PSP) will be generated. One can use certain time course to depict this dynamic PSP change. In leaky integrate-and-fire model, the post-synaptic potential will gradually decrease if no spikes received since last received spike. Therefore, in order to generate a post-synaptic spike, this post-synaptic neuron needs to receive lots of spikes within a relative small time window so that its PSP can reach the predefined threshold. The dynamic procedure of leaky integrate-and-fire model can be summarized as follows: when a post-synaptic neuron receives presynaptic spikes, it will generate dynamic synaptic current and this dynamic current will thus produce dynamic synaptic voltage. A post-synaptic spike will fired if the dynamic synaptic voltage reaches the predefined post-synaptic potential threshold. The dynamic post-synaptic current can be expressed as follows:

$$I_i(t) = \sum_j w_{ij} \sum_f \alpha(t - t_j^{(f)}) \quad (5)$$

where $t_j^{(f)}$ represents the time of the f -th spike of the j -th presynaptic neuron; w_{ij} is the strength of the synaptic efficacy between neuron i and neuron j . $\alpha(t)$ is the time course function, which can be expressed as follows:

$$\alpha(t) = \frac{1}{\tau} \exp\left(-\frac{t}{\tau}\right) \Theta(t) \quad (6)$$

where Θ is the Heaviside step function with $\Theta(t) = 1$ for $t > 0$ and $\Theta(t) = 0$ else. τ is the time constant. For a given time-varying input current $I(t)$, the dynamic voltage $V(t)$ can be computed as follows:

$$V(t) = V_r \exp\left(-\frac{t-t_0}{\tau_m}\right) + \frac{R}{\tau_m} \int_0^{t-t_0} \exp\left(-\frac{s}{\tau_m}\right) I(t-s) ds \quad (7)$$

where the initial condition $V(t_0) = V_r$ and τ_m is the membrane time constant. R represents the resistance. This equation describes the dynamics of the

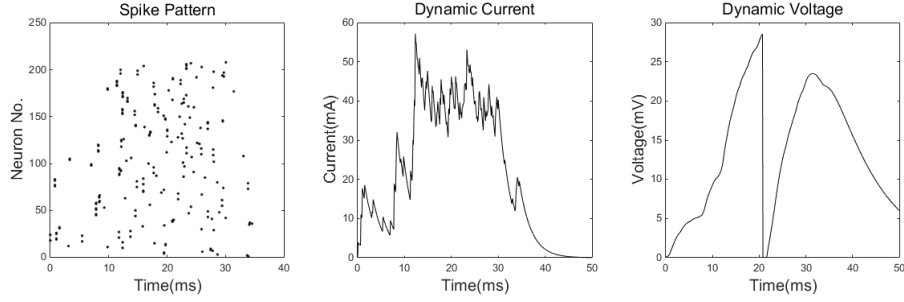


Figure 8: Schematic diagram of leaky integrate-and-fire model.

membrane potential between successive spiking events. When the membrane potential reaches the threshold, it will fire a spike, followed by the absolute refractory period (resets to V_r) and then start to evolve afterwards.

In this paper, a dynamic post-synaptic potential threshold has been proposed in the training period. For each input image, we do not set post-synaptic potential threshold for the first run and collect the generated dynamic voltage due to the input spike pattern. And then the maximum value of the dynamic voltage needs to be found after collecting all dynamic voltage within the predefined time window. Finally, the associated post-synaptic potential threshold has been set to a percentage of this maximum value. By doing this, each input image can be ensured to be trained during the learning procedure. Such scenarios with only a little part of training samples have been actually used (especially those training samples with relatively large intra-class variance) will be avoided. Each input spike pattern will contribute its part to the final learning efficacy matrix with certain selectivity.

$$V_{thr} = k * \max(V(t)) \quad (8)$$

where V_{thr} is the post-synaptic potential threshold and $V(t)$ represents dynamic voltage. $\max(V(t))$ is the maximum value of dynamic voltage within the predefined spiking time window and k (0.8 in this paper) depicts a positive constant within the range $[0, 1]$.

Fig.8 uses the same input image as Fig.7 and shows its spike pattern, dy-

dynamic current and dynamic voltage. From Fig.8, it can be seen that the oscillation of dynamic current depends on the closeness of the local spike packages. When the post-synaptic neuron receives lots of spikes from presynaptic neurons in a short time window, the dynamic current will increase dramatically and then gradually decline if no spikes received afterwards. This dynamic current will generate dynamic voltage (membrane potential) in the post-synaptic neuron. When the dynamic voltage reaches its predefined post-synaptic potential threshold, the post-synaptic neuron will fire a spike, followed by a quite short refractory period (about 1ms) and then start integrating again. What's more, by using leaky integrate-and-fire model, only spikes within a short time window can stimulate the post-synaptic neuron to fire a post-synaptic spike. Those presynaptic spikes with much later or earlier have no influence on the procedure of generating a post-synaptic spike.

3.2. STDP Learning Rule

Hebb's postulate [12] may be the most important theory in neuroscience trying to explain the adaptation of neurons in the brain during the learning process. It can be summarized as "Cells that fire together, wire together". This kind of statement emphasizes the causality between pre- and postsynaptic neurons. Hebb emphasized that cell A needs to take part in firing cell B, and such causality can only occur if cell A fires just before, not at the same time as, cell B.

Spike timing dependent plasticity (STDP) [7],[8],[9],[10],[11] has been proved to be a quite effective learning rule by neuroscientists, which adjusts the efficacy of synaptic connections based on the relative timing of post-synaptic spike and its input presynaptic spike. Like Hebb's postulate, it also emphasizes the causality between pre- and postsynaptic neurons. Actually, it can be considered as a temporally asymmetric form of Hebb's rule. In neuroscience, long-term potentiation (LTP) is a persistent strengthening of synapses based on recent patterns of activity, while long-term depression (LTD) is an activity-dependent long-lasting reduction in the efficacy of neural synapses. When a presynaptic

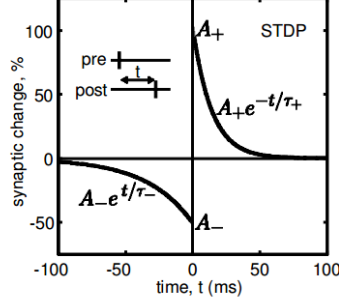


Figure 9: STDP learning window.

spike fires slightly earlier than the post-synaptic spike, the associated synaptic efficacy will be potentiated (LTP). While the associated synaptic efficacy will be depressed (LTD) if the presynaptic synaptic spike fires later than the post-synaptic spike. The STDP function $W(t)$ can be expressed as follows (t is the time difference between pre- and postsynaptic spikes):

$$W(t) = A_+ \exp\left(-\frac{t}{\tau_+}\right) \quad \text{for } t > 0 \quad (9)$$

$$W(t) = -A_- \exp\left(\frac{t}{\tau_-}\right) \quad \text{for } t < 0 \quad (10)$$

where A_+ and A_- represent amplitude of LTP part and LTD part of the *learning window*, respectively. τ_+ and τ_- are time constant for LTP and LTD, respectively. For biological reasons, it is desirable to keep the synaptic efficacy in a predefined range. Thus, a soft bound strategy [32],[33] has been used to ensure the synaptic efficacy remains in the desired range $w^{min} < w_j < w^{max}$, here, w^{min} and w^{max} represent minimum and maximum value, respectively. The soft bound strategy can be expressed as follows (for the sake of simplicity, the lower bound is set to zero in most models):

$$A_+(w_j) = (w^{max} - w_j)\eta_+ \quad \text{and} \quad A_-(w_j) = w_j\eta_- \quad (11)$$

315 where η_+ and η_- are positive constants. Fig.9 shows one example of STDP learning window.

To model a biological system, it is desirable to keep the synaptic efficiency

in a predefined range, for example between $[0, 1]$. During real implementation, for the sake of simplicity, the minimum value of the synaptic weight w^{min} is set to zero in this paper. Moreover, the $C1$ features within the feature extracting layer have been normalized to $[0, 1]$. To compute the similarity between the input features and the corresponding synaptic efficiencies, the maximum value of the synaptic weight w^{max} is set to 1 in this paper.

Besides the soft bound strategy, there are two ways of spike-interaction within STDP learning rule: temporal all-to-all and nearest-neighbor spike-interaction. In Eq.(9) and Eq.(10), the first one sums over all presynaptic spike arrivals and all post-synaptic spikes, while the latter one restricts the interaction so that only nearest spikes interact. In this paper, after several experimental tests, we choose the first one in our method as it can achieve much higher performance compared with the latter one.

STDP is an unsupervised learning rule, which requires no prior information or teaching signals. It can adaptively find the desired spike pattern when the synaptic efficacy matrix remains stabilized. It has been proven that STDP can reliably find the start of repeating pattern even there are spike jitters or spontaneously activities existed [6]. In order to reach stable status for synaptic efficacy, the predefined post-synaptic potential threshold needs to be tuned around its optimum value. If the threshold takes the maximum threshold value, only the exact same input can generate a post-synaptic spike. It is impossible for the test samples to be exact same as training sample in real world. Otherwise, if the threshold takes value around the minimum threshold value, the noise spike pattern will mostly be potentiated and thus the wanted spike pattern will be ignored [31]. In this paper, the proposed dynamic threshold method has been used to choose the optimum dynamic voltage threshold.

4. Experimental Results and Analysis

In order to validate the proposed spike timing-based feed-forward spiking neural network and its unsupervised STDP learning rule, we will use MNIST

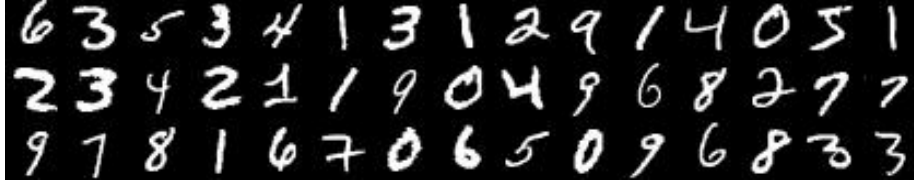


Figure 10: Random examples of MNIST database.

handwritten digital characters database as the training and testing database. Furthermore, parameter settings of the experiment will be elaborated in details, followed by the according experiments and discussions. Discussion will be
350 provided in the final subsection.

4.1. MNIST database

MNIST handwritten digital characters database [34] is a well known benchmark in pattern recognition field. It contains 60000 training samples and 10000 testing samples (all sample size is 28×28). It includes 10 classes that is dig-
355 ital handwritten digits from 0 to 9. Fig.10 shows some examples of MNIST database. It can be seen that the database has large intra-class variance, which could be a real challenge for the proposed method. For instance, the digit 1 and 7 in Fig.10 have different external shape (the fifth digit in the second row and the sixth digit in the last row have significant different external shape compared with other samples in their class). Sometimes, even human being cannot
360 easily recognize some digits of the database. For example, the fifth digit in the last row could be seen as 4 or 6 and each one can have their own opinion. Therefore, by testing the performance using this MNIST handwritten digital characters database, one can conclude the advantages and limitations of the
365 proposed SNN and its own unsupervised learning method.

4.2. Parameter Setting

Before elaborating the experimental results, the experimental parameter settings using in the SNN is needed to state first. The time resolution of this experiment is 0.1ms. In this paper, for the S1 and C1 features of the feature

370 extracting layer, we used the same parameter setting as the HMAX model. The
 only different is that only first 2 scales are used in the experiment. For spiking
 encoding layer, we use a linear equation and the specific absolutely C1 feature
 values to generate the spikes. For the leaky integrate-and-fire model used in
 output layer, the α function described in equation (6) has been used to mimic
 375 the time course of the dynamic synaptic current and the time constant τ used
 in the equation is set to 2.5ms. Equation (7) was used to compute the dynamic
 synaptic voltage, where the initial condition V_r is 0, the membrane time con-
 stant is 10 ms and the resistance R is 0.1 m Ω . The absolute refractory period
 is set to 1 ms. For STDP learning of the output layer, the time constant τ_+
 380 and τ_- are set to 0.0168 and 0.0337, respectively. For soft bound strategy, the
 maximum weight and the minimum weight are set to 1 and 0, respectively. The
 positive constants η_+ and η_- are set to 0.03125 and 0.0265625, respectively.

To increase the level of realism in a neural simulation, spiking neural net-
 work (SNN) is often used as the neural network model. Moreover, within SNN,
 385 different parameters are required to regulate different dynamic procedures, such
 as generating dynamic synaptic current, obtaining the dynamic synaptic volt-
 age, spike-timing dependent plasticity (STDP) learning procedure. To simulate
 a biologically realistic learning procedure, the range of allowable values for each
 parameter is often fixed. For instance, the membrane time constant is often
 390 chosen from a range of 10 – 20mV, the absolute refractory period is often set
 to a range of 1 – 3ms. Thus, given the predefined ranges, different evolutionary
 algorithms (EA) can be used to tune the parameters to their optimum val-
 ues. Specifically, to achieve the best classification performance, the parameter
 settings described in section 4.2 have been used in this paper.

395 4.3. Experiments and Discussions

Ideally, we expect that there are little intra-variance or even no intra-variance
 existed in the extracted features. However, such critical need cannot easily be
 satisfied. In the following subsections, we will discuss how STDP learning can
 handle the scenarios with no intra-variance and large intra-variance, respec-

400 tively.

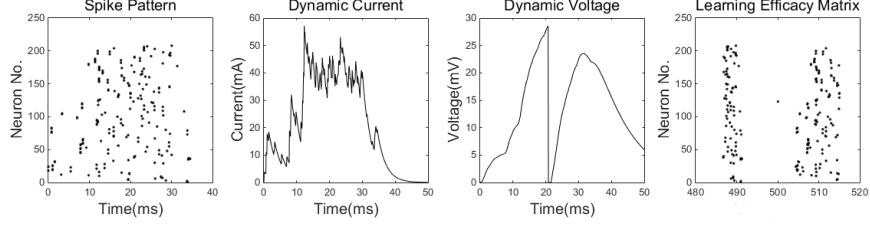
4.3.1. STDP Learning with no Intra-class Variance

Fig.11 shows dynamic learning procedure of generating selectivity after using unsupervised STDP learning method. Fig.11 (a) shows the beginning of the learning procedure. It can be seen that the dynamic synaptic current fluctuates over the whole time window and the synaptic voltage reaches its threshold at about 21ms. The synaptic efficacy weights are relatively random at this stage. After presenting the same input image (same input image in Fig.7) to the SNN system about 300 times, the selectivity finally emerged, just as the Fig.11 (b) shows. At this stage, the synaptic current only fluctuates over the first half time window and the synaptic voltage fires the spike at about 13ms. What's more, the synaptic efficacy matrix has a special status with most weights take 0 and the rest take 1 [35], [36]. Therefore, the selectivity to this specific input image emerges. However, such learning results can be generated only if the intra-class variance of the input images remains at a reasonable level.

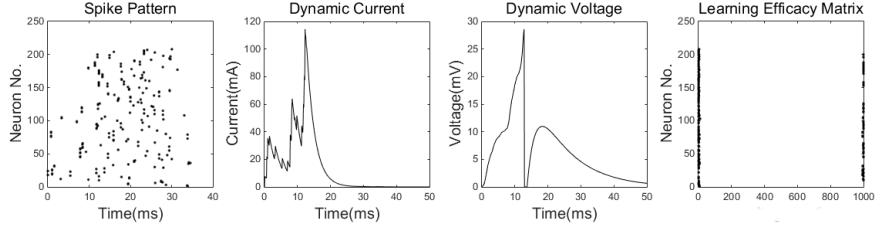
415 4.3.2. STDP Learning with Relatively Large Intra-class Variance

In Fig.11, an ideal experimental condition that the input image with no intra-class variance has been tested with the proposed timing-based feed-forward spiking neural network and obtained an ideal STDP learning efficacy matrix. However, in real world, such ideal condition is hardly achieved as vast intra-class variance existed among the samples. In this paper, we proved that certain selectivity to the input can be learned using unsupervised STDP learning rule even If the intra-class variance level of the input remains at a relatively high level, as shown in fig.12 (a),(b),(e),(f).

In Fig.12, two groups with two input images have been used to learn the selectivity using unsupervised STDP learning rule. In fact, input images (a) and (e) are the same. The group one uses (a) and (b) as its input images, (c) and (d) represent dynamic efficacy matrix of 20-th and 200-th iterations, respectively. (e) and (f) have been fed into group two, and thus obtained its



(a) Status of the first iteration of STDP learning



(b) Status of the 200-th iteration of STDP learning

Figure 11: Generating selectivity by using unsupervised STDP learning. The resolution for learning efficacy matrix (x axis) is set to 0.001 so that the maximum value equals to 1.

20-th and 200-th dynamic efficacy matrix, shown as (g) and (h). Here, one
430 iteration means sequentially fed the two input images into the feed-forward
spiking neural network one by one. From Fig.12, one can easily concluded
that training samples with more intra-class variance will somehow hard to learn
the selectivity. In other words, the dynamic efficacy matrix can be very hard
to concentrate on the extreme values of 0 and 1 if having high level of intra-
435 class variance within the training samples. Compared to the same input image
(a), input image (f) is much more different than the input image (b), thus the
dynamic efficacy matrix of group two had more weights lingering between the
extreme values of 0 and 1.

As mentioned in the above section, a dynamic voltage threshold strategy
440 has been proposed to guarantee each training sample will be properly learned.
Table.I shows the correct classification comparison with the proposed dynamic
voltage threshold and the predefined voltage threshold. Notably, the predefined
voltage thresholds have been set within a certain range (10-30 mV) around its

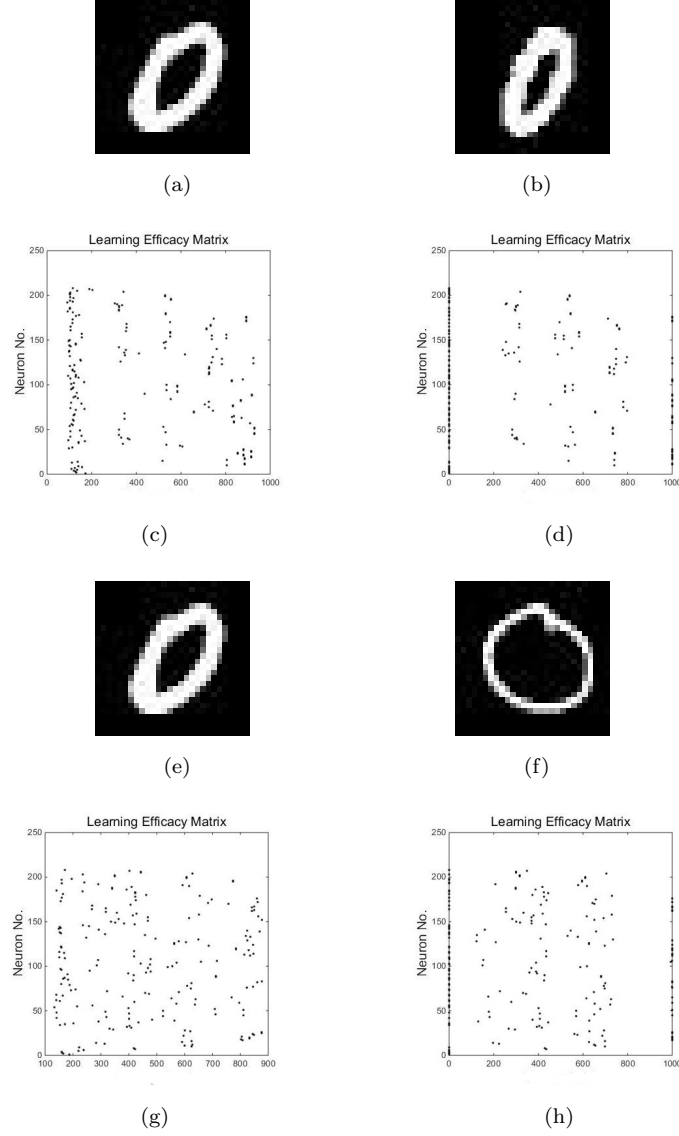


Figure 12: (a) and (b) are input images of group one, (c) and (d) are dynamic efficacy matrix of 20-th and 200-th iterations, respectively; (e) and (f) are input images of group two, (g) and (h) are dynamic efficacy matrix of 20-th and 200-th iterations, respectively. The resolution for learning efficacy matrix (x axis) is set to 0.001 so that the maximum value equals to 1.

Table 1: Impact of dynamic voltage threshold on recognition rate.

Random test	With dynamic threshold	With predefined threshold (mV)		
		10	20	30
1	0.79	0.71	0.77	0.75
2	0.82	0.75	0.81	0.68

- Note: 0.79 in this table means 79% correct classification rate.

optimum value (20 mV). It can be seen that the dynamic voltage threshold
445 strategy can not only ensure learning each training sample properly but also
generate the best correct classification performance.

Fig.13 shows the dynamic learning procedure using the proposed SNN and
its STDP learning rule for one class. It is worth to note that one iteration
in this experiment means sequentially feed 50 different training samples within
450 certain class one by one. From Fig.13, it can be seen that the dynamic status
only have a very limited changes. However, even the intra-class variance in
the experiment remains at a relatively high level, the training samples are not
totally independent (e.g. totally random samples), and thus such seemingly
random learning efficacy matrix may contains certain selectivity to the input.

455 One question still needs to answer - to achieve the optimal performance, how
many iterations should the STDP learning method run? We will answer this
question in the following subsections with carefully designed experiments.

4.3.3. Experiments on MNIST Database

There are total 60,000 training samples in MNIST database, as mentioned
460 above, given a real-time learning circumstance, it is hard to fully exploit the
whole database with limited time. We will use a cross validating method to
test the proposed algorithm and to answer the above question. Cross validating
method, which randomly selects limited samples for training and testing, creates

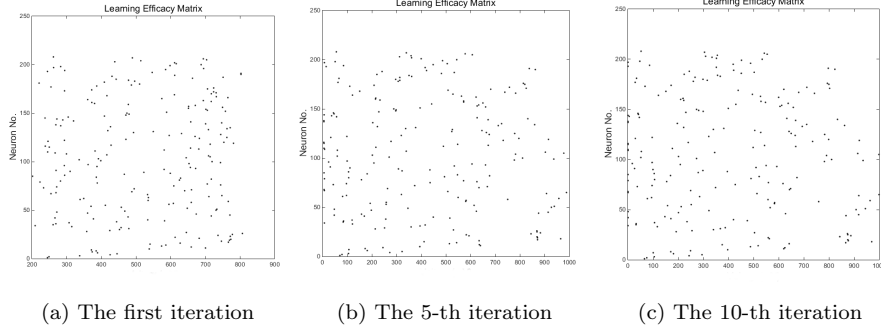


Figure 13: Learning efficacy matrix with large intra-class variance. The resolution for learning efficacy matrix (x axis) is set to 0.001 so that the maximum value equals to 1.

a scenario most similar to a real-time learning situation.

465 From the MNIST, we randomly choose 50 different training samples for each class and 100 different testing samples to test the correct classification rate. In the following experiments, each test follows the same procedure mentioned above. For fair comparison, each iteration within each test uses the same randomly chosen training samples and testing samples.

470 Table.II shows the corresponding correct classification rate performance when using the experimental conditions mentioned above. Average correct classification rate also has been added in the table. It can be seen that, with one iteration only, almost all the tests achieved the highest performance. This suggests that the proposed learning method is suitable for real-time learning.

475 Fig.14 shows standard error performance using different iterations. It can be seen that, along with increasing of iterations, the correct classification rate gradually decreases. Tests with one iteration only seems to convey the least standard error. Such characteristic indicates the learning methods with one iteration only are more reliable than that with more iterations.

480 Why more iterations have not led to better performance in this case? This is because, for precisely timed spikes (meaning small temporal jitter of the pre- and postsynaptic spikes), the synaptic weight will be tuned according to STDP learning window. Specifically, the synaptic weight saturates close to

Table 2: Cross validation with different iterations in 10 random tests.

Tests	Number of iterations				
	1	2	3	4	5
1	0.81	0.85	0.81	0.76	0.71
2	0.83	0.78	0.75	0.72	0.74
3	0.81	0.87	0.88	0.87	0.87
4	0.8	0.81	0.81	0.78	0.76
5	0.84	0.8	0.8	0.78	0.77
6	0.8	0.78	0.74	0.73	0.72
7	0.81	0.8	0.78	0.77	0.75
8	0.84	0.81	0.78	0.8	0.79
9	0.81	0.79	0.74	0.74	0.73
10	0.84	0.79	0.78	0.79	0.79
Average	0.819	0.808	0.787	0.774	0.763

• Note: 0.81 in this table means 81% correct classification rate.

its maximum value if the presynaptic spikes arrived before the postsynaptic
485 neuron is firing. If the timing is the other way round, the synaptic weight
will be approximately zero. Therefore, for precisely timed spikes, the result
performance will become better if increasing the learning iterations.

However, if the temporal jitter of the pre- and postsynaptic spikes escalated,

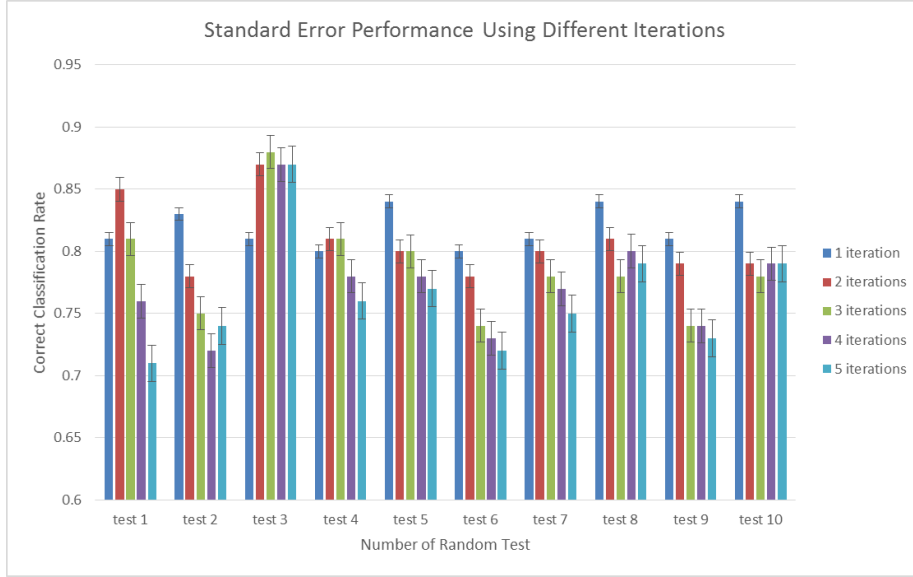


Figure 14: Standard error performance using different iterations.

the synaptic weight will take an intermediate value determined by non-Hebbian
 490 terms rather than STDP learning window [37]. In this paper, we choose $C1$
 features instead of more abstracted features to get the best balance between
 learning speed and performance. Since the generated $C1$ features within the
 proposed SNN still contain relatively large intra-class variances (means relatively
 large temporal jitter of pre- and postsynaptic spikes), increasing the iteration
 495 times implies the level of the temporal jitter of pre- and postsynaptic spikes is
 increased, which may lead to the poor performance with more iterations.

STDP learning will generate the perfect learning efficacy matrix if there are
 no intra-class variance or the intra-class variance remains at a relatively low
 level, which indicates the extracted features should have a relatively high level
 500 invariance. However, in real world, such strict demand can hardly be achieved.
 In most cases, those extracted features will still contain relatively large intra-
 class variance. Therefore, it is desirable to find a reasonable learning method
 to deal with those features with relatively large variance, and still obtain a
 satisfactory performance in the end.

Table 3: Performance comparison of three methods(%).

Method	Correct rate	Wrong rate	Unknown rate
The proposed method	82 ± 2	18 ± 2	0
Tempotron rule [14]	78.5 ± 1.85	18.35 ± 1.85	3.15 ± 1.64
SVM [14]	79.33 ± 2.03	18.15 ± 1.69	2.53 ± 2.04

505 In paper [14], the authors used a supervised temporal learning rule (named Tempotron Rule) to train the MNIST database (almost same experimental conditions as this paper) and achieved 79% correct classification rate in the end. Unlike this state-of-art learning method, the proposed algorithm uses unsupervised STDP learning rule with dynamic post-synaptic potential threshold during
510 the learning procedure. Dynamic post-synaptic potential threshold guarantees that each training sample will be properly learned.

What's more, since the proposed method uses a more pervasive unsupervised learning rule, no prior information or teaching signals are needed in this paper. Table.III and Fig.15 show the final classification performance comparison
515 of three different methods. It can be seen that the unknown rate of the proposed method is 0, which means each testing sample would be recognized as one possible class. Compared with Tempotron Rule, the proposed method achieves better correct rate at around 82%, while still remains slightly less wrong rate. Finally, Table.IV shows the speed test results for the training and testing periods respectively. It can be seen that the learning and testing speeds are quite
520 impressive - 21.3 fps in training and 17.9 fps in testing, both can be operating in real time.

In real world, there are at least two critical restrictions determining a learning methodology. One is time limit, often there is not enough time to learn,
525 especially in real time scenarios. Second restriction is the size of data can be ac-

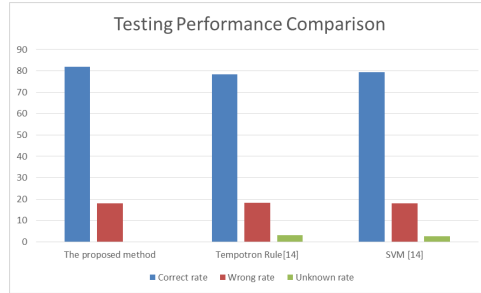


Figure 15: Performance comparison of three methods.

cessed given time restriction - for example, a learning algorithm can only access to a few data points or encounter a few events. Our proposed method performs much better when the number of iteration and data size are both restricted - a situation very similar to that a human brain faces.

530 For those methods that may have achieved better performance on the MNIST database, all of them have assumed that they have enough time to learn all the database with unlimited iterations before carry out a test. This is impossible in real world. In any cases of a real world situation, new visual pattern should be learned in a limited time window to adapt to new environments or changes
535 promptly. Moreover, given a real-time learning circumstance, it is hard to fully exploit the whole database with limited time. Therefore, to enable a machine to learn visual patterns in real time proposes a huge challenge - the algorithms underlying should process a large volume of visual data in an extremely short period of time, which has been proved difficult. However, it is also obvious that
540 a biological brain could cope with these large volume of visual data effortlessly in real time. Inspired by this phenomenon, we propose a novel method taking advantage of spiking neural network (SNN) and spiking timing-based coding scheme. Moreover, an unsupervised STDP learning method is employed in the proposed method to make it more practical. Fast cross-validated experiments
545 using MNIST database are carried out to prove the efficiency and accuracy of the proposed method.

Table 4: Running speed tests.

Item	Running time(s)	Equivalent to frames per second
Training	23.32	21.3
Testing	5.63	17.9

- Note: the above results are the mean value of 10 random tests gathered from a laptop with Intel 3rd Gen Core at 2.5 GHz, 8G RAM and 128G SSD. The whole training procedure includes 500 frames/samples represent total 10 classes (50 samples for each class) and the whole testing procedure includes 100 samples.

4.4. Discussion

From a broader perspective, the bio-inspired visual pattern recognition or similar applications can be considered as simulating the complicated biological processing mechanism. Many researches have been deployed to mimic the complicated processing procedure of the mammalian brain. In paper [38], the authors proposed a synaptic weight association training (SWAT) algorithm for spiking neural networks, which merges the Bienenstock-Cooper-Munro (BCM) learning rule [39] with spike timing dependent plasticity (STDP) [40]. Such rule yields a unimodal weight distribution where the height of the plasticity window associated with STDP is modulated causing stability after a period of training. Other papers [41], [42], [43], [44] combines the self-organizing map (SOM), a neural network algorithm to create topographically ordered spatial representations of an input data set using unsupervised learning, with spiking neural network to mimic the feature maps found in mammalian cortices. Moreover, the authors in [45] propose a novel unsupervised approach for exemplar extraction through structured sparse learning, which not only considers the reconstruction capability and the sparsity, but also the diversity and robustness. The effec-

tiveness of this method has been verified by experiments of various examples
565 including traffic sign sequences. Within visual cortex, it is known that neurons
are silent most of the time, firing only occasionally (sparse firing). Sparse firing
is a hallmark of neurons at almost all stages of visual cortex. Like those meth-
ods mentioned above, the proposed spiking neural network and its unsupervised
STDP learning method can be considered as another try for simulating the pro-
570 cessing procedure of the mammalian’s brain. Besides mimicking mammalian
brain, many researchers pay their attention to those lower organisms, such as
locust or fly [46], [47], [48], [49]. However, such methods did not transform the
analog input data into spiking patterns and thus lack the advantages of the
spiking neural network.

575 Real-time learning proposed significant challenge to machine learning meth-
ods with limited processing time and huge data sets. Inspired by biological
visual brain, the spike pattern generated by temporal spike coding conveys sig-
nificant spatio-temporal information about the input data, which can be used
for the visual pattern recognition task, as demonstrated above. It is also impor-
580 tant to note that STDP can not only used in feature extracting layers to learn
the connection between C1 S2 ([13]), it can also be used in learning spiking
patterns after spiking encoding layer as demonstrated in this study. Using only
the C1 and S1 for feature extraction, avoiding complicated and time consuming
high level feature extraction, a spike timing-based SNN can achieve acceptable
585 recognition rate after fast STDP learning. Feature extraction mechanisms in
human or animal’s brain remains active research field in both neuroscience and
computer vision. It can be predicted that with more natural feature extrac-
tion schemes revealed and integrated into the SNN structure in the future, the
recognition rate and learning efficiency will be further improved.

590 5. Conclusion

Real-time learning needs algorithms operating in a fast speed comparable to
human or animal, however this is a huge challenge in processing visual inputs

at milliseconds scale. In the above chapters, we proposed a novel real-time learning method by combining the spike timing-based feed-forward spiking neural network (SNN) and the fast unsupervised spike timing dependent plasticity learning method with dynamic post-synaptic thresholds. Fast cross-validated experiments using MNIST database showed the high efficiency of the proposed method at an acceptable accuracy. Our research may also add to the further understanding of the dynamic processing procedure existed in brain’s ventral stream.

Acknowledgements

The authors have been supported by EU FP7 project LIVCODE(295151), HAZCEPT(318907), EU Horizon 2020 project ENRICHME(643691) and STEP2DYNA(691154).

References

- [1] F. Rieke, R. Warland, D. de Ruyter van Steveninck, W. Bialek, Spikes: Exploring the neural code, MIT Press, Cambridge, MA, 1996.
- [2] M. N. Shadlen, W. T. Newsome, Noise, neural codes and cortical organization, *Curr. Opin. Neurobiol.* 4 (1994) 569–579.
- [3] J. M. Brader, W. Senn, S. Fusi, Learning Real-World Stimuli in a Neural Network with Spike-Driven Synaptic Dynamics, *Neural Computation* 19 (11) (2007) 2881–2912.
- [4] P. U. Diehl, M. Cook, Unsupervised learning of digit recognition using spike-timing-dependent plasticity, *Frontiers in Computational Neuroscience* (2015) 99.
- [5] S. Thorpe, D. Fize, C. Marlot, Speed of processing in the human visual system, *Nature* 381 (6582) (1996) 520–522.

- [6] T. Masquelier, R. Guyonneau, S. J. Thorpe, Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains, PLoS ONE 3 (1) (2008) e1377.
- [7] W. Gerstner, R. Kempter, J. L. van Hemmen, H. Wagner, A neuronal learning rule for sub-millisecond temporal coding, Nature 386 (1996) 76–78.
- [8] K. Kitano, H. Cateau, T. Fukai, Sustained activity with low firing rate in a recurrent network regulated by spike-timing-dependent plasticity, Neurocomputing 4446 (2002) 473–478.
- [9] H. Markram, J. Lubke, M. Frotscher, B. Sakmann, Regulation of synaptic efficacy by coincidence of postsynaptic apss and epsps, Science 275 (1997) 213–5.
- [10] F. Henry, E. Dauc, H. Soula, Temporal pattern identification using spike-timing dependent plasticity, Neurocomputing 70 (1012) (2007) 2009–2016.
- [11] A. Shahim-Aeen, G. Karimi, Triplet-based spike timing dependent plasticity (TSTDTP) modeling using VHDL-AMS, Neurocomputing 149, Part C (2015) 1440–1444.
- [12] D. O. Hebb, The Organization of Behavior: a neuropsychological theory, Wiley, New York, 1949.
- [13] T. Masquelier, S. J. Thorpe, Unsupervised learning of visual features through spike timing dependent plasticity, PLoS computational biology 3 (2) (2007) e31.
- [14] Q. Yu, H. Tang, K. Tan, H. Li, Rapid feedforward computation by temporal encoding and learning with spiking neurons, IEEE Trans. Neural Networks and Learning Systems 24 (10) (2013) 1539–1552.
- [15] S. Thomas, W. Lior, B. Stanley, R. Maximilian, P. Tomaso, Robust object recognition with cortex-like mechanisms, IEEE Trans. Pattern Anal. Mach. Intell. 29 (3) (2007) 411–426.

- [16] Y. Bengio, Learning deep architectures for ai, *Found. Trends Mach. Learn.* 2 (1) (2009) 1–127.
- [17] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (8) (2013) 1798–1828.
- [18] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, T. Poggio, A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex.
- [19] T. Christian, T. Nicolas, C. Matthieu, Extended coding and pooling in the hmax model, *IEEE Trans. Image Processing* 22 (2) (2013) 764–777.
- [20] W. Gerstner, Population dynamics of spiking neurons: fast transients, asynchronous states and locking, *Neural Comput.* 12 (2000) 43–89.
- [21] N. Brunel, F. Chance, N. Fourcaud, L. F. Abbott, Effects of synaptic noise and filtering on the frequency response of spiking neurons, *Phys. Rev. Lett.* 86 (2001) 2186–2189.
- [22] S. J. Thorpe, Spike arrival times: a highly efficient coding scheme for neural networks, *Parallel processing in neural systems and computers*, North-Holland. (1990) 91–94.
- [23] W. C. Butts, D. A., J. J. et al., Temporal precision in the neural code and the timescales of natural vision, *Nature* 449 (7158) (2007) 92–95.
- [24] R. B. Stein, E. R. Gossen, K. E. Jones, Neuronal variability: noise or part of the signal?, *Nat. Rev. Neurosci.* 6 (5) (2005) 389–397.
- [25] S. Wu, S. Amari, H. Nakahara, Population coding and decoding in a neural field: a computational study, *Neural Comput.* 14 (5) (2002) 999–1026.
- [26] S. Bohte, H. La Poutre, J. Kok, Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks, *IEEE Transactions on Neural Networks* 13 (2) (2002) 426–435.

- [27] S. Thorpe, D. Fize, C. Marlot, Speed of processing in the human visual system, *Nature* 381 (1996) 520–522.
- 675 [28] A. Delorme, J. Gautrais, R. van Rullen, S. Thorpe, Spikenet: a simulator for modeling large networks of integrate and fire neurons, *Neurocomputing* 26 (1999) 989–996.
- [29] A. Delorme, L. Perrinet, S. Thorpe, Networks of integrate-and-fire neurons using rank order coding, *Neurocomputing* 38-40 (2001) 539–545.
- 680 [30] A. Delorme, S. Thorpe, Face identification using one spike per neuron: resistance to image degradation, *Neural Networks* 14 (6-7) (2001) 795–803.
- [31] W. Gerstner, W. M. Kistler, *Spiking neural models*, Cambridge University Press, Cambridge, MA, 2002.
- [32] M. C. W. van Rossum, G. Q. Bi, G. G. Turrigiano, Stable hebbian learning from spike time dependent plasticity, *Journal of Neuroscience* 20 (88) (2000) 12–21.
- 685 [33] J. Rubin, R. Gerkin, G. Bi, C. Chow, Calcium time course as a signal for spike-timing-dependent plasticity, *J Neurophysiol* 93 (2005) 2600–13.
- [34] <http://yann.lecun.com/exdb/mnist/>, accessed: 2013-09-30.
- 690 [35] R. Guyonneau, R. VanRullen, S. J. Thorpe, Neurons tune to the earliest spikes through STDP, *Neural Computation* 17 (4) (2005) 859–879.
- [36] R. Guyonneau, R. Vanrullen, S. J. Thorpe, Temporal codes and sparse representations: a key to understanding rapid processing in the visual system, *Journal of Physiology, Paris* 98 (4-6) (2004) 487–497.
- 695 [37] W. M. Kistler, J. L. van Hemmen, Modeling synaptic plasticity in conjunction with the timing of pre- and postsynaptic action potentials, *Neural Computation* 12 (2) (2000) 385–405.

- [38] J. Wade, L. McDaid, J. Santos, H. Sayers, SWAT: A Spiking Neural Network Training Algorithm for Classification Problems, *IEEE Transactions on Neural Networks* 21 (11) (2010) 1817–1830.
- [39] E. L. Bienenstock, L. N. Cooper, P. W. Munro, Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex, *Journal of Neuroscience* 2 (1982) 32–48.
- [40] D. Liu, S. Yue, Visual pattern recognition using unsupervised spike timing dependent plasticity learning, in: 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 285–292.
- [41] B. Ruf, M. Schmitt, Self-organization of spiking neurons using action potential timing, *IEEE Transactions on Neural Networks* 9 (3) (1998) 575–578.
- [42] N. Manukyan, M. Eppstein, D. Rizzo, Data-Driven Cluster Reinforcement and Visualization in Sparsely-Matched Self-Organizing Maps, *IEEE Transactions on Neural Networks and Learning Systems* 23 (5) (2012) 846–852.
- [43] C.-C. Hsu, S.-H. Lin, Visualized Analysis of Mixed Numeric and Categorical Data Via Extended Self-Organizing Map, *IEEE Transactions on Neural Networks and Learning Systems* 23 (1) (2012) 72–86.
- [44] T. Rumbell, S. Denham, T. Wennekers, A Spiking Self-Organizing Map Combining STDP, Oscillations, and Continuous Learning, *IEEE Transactions on Neural Networks and Learning Systems* 25 (5) (2014) 894–907.
- [45] H. Liu, Y. Liu, F. Sun, Robust exemplar extraction using structured sparse coding, *IEEE transactions on neural networks and learning systems* 26 (8) (2015) 1816–1821.
- [46] F. C. Rind, D. I. Bramwell, Neural network based on the input organization of an identified neuron signaling impending collision, *Journal of Neurophysiology* 75 (3) (1996) 967–985.

- [47] S. Yue, F. Rind, Collision detection in complex dynamic scenes using an LGMD-based visual neural network with feature enhancement, *IEEE Transactions on Neural Networks* 17 (3) (2006) 705–716.
- [48] S. Yue, F. Rind, Redundant Neural Vision Systems -Competing for Collision Recognition Roles, *IEEE Transactions on Autonomous Mental Development* 5 (2) (2013) 173–186.
- [49] B. Hu, S. Yue, Z. Zhang, A rotational motion perception neural network based on asymmetric spatiotemporal visual information processing, *IEEE Transactions on Neural Networks and Learning Systems* PP (99) (2016) 1–19.